ARMY RESEARCH LABORATORY

Spreadsheet Proof of Arguments

by John D. Sullivan

ARL-TR-2815

September 2002

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-2815 **September 2002**

Spreadsheet Proof of Arguments

John D. Sullivan
Weapons and Materials Research Directorate, ARL

Abstract

This report describes the use of a spreadsheet to solve simple problems in prepositional logic. A spreadsheet can easily generate and recall truth tables, and with its built-in logical functions and connectives it can calculate the truth value of logic expressions. Expressions can also be premises and together with a conclusion constitute an argument, which can be shown valid with a truth table.

This report covers the application of a spreadsheet to truth table generation, evaluation of logical expressions, recasting arguments into spreadsheet form, and demonstration of validity.

Acknowledgments

The author thanks Timothy J. Mermagen of the Individual Performance Branch, Soldier Performance Division, Human Research and Engineering Division, U.S. Army Research Laboratory, for critically reading this report.

INTENTIONALLY LEFT BLANK.

Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

1. Introduction

From the standpoint of logic, the problem with any argument is to determine that it is valid. By abstracting an argument from (English) language to a symbolized language, eye clutter is much reduced and the tools of logic can be applied to investigating the validity of the argument. Symbolization takes skill separate from the method of solution used. Getting solutions to logic problems may be done by truth table or deductive methods. The truth table is the basic introductory method of logic. Deduction is a much shorter method of validating arguments and the only method for complicated arguments.

Both methods have drawbacks. Truth tables have a conventional order, which the user may not know. Working with even small truth tables is tedious and error prone if done by hand, and it is impractical for most sizes. Within arguments, it is restricted to the least complex ones that have sentences of a simple variety—subject, verb, and object. The deductive method is much shorter, but is artful in nature and requires rule learning and skill in symbol manipulation. Still, it is by far the normal method of validating arguments.

For logic problems that are amenable to the truth table method, a spreadsheet is a moderately easy and very accurate tool to achieve the solution. However, the spreadsheet approach is infeasible without having connectives like *and, or,* or *not,* to operate on symbolized sentences. But these connectives are logical operators that do reside in every spreadsheet program, for other reasons, and are here given an unintended new use.

A method of generating truth tables via spreadsheets is shown. Proving validity of arguments with symbolic logic is recounted. Intertwined are examples of problems susceptible to the truth table method, such as finding the truth values of logic expressions and determining validity of simple arguments. The emphasis is on the spreadsheet use of truth tables, but for contrast the deductive method is also used.

2. Truth Table Generation

A logical variable takes on the values of *true* (T) or *false* (F). A truth table with several variables is an orderly list of all possible arrangements of T and F. The size of a table of *n* variables is 2^n rows by n columns. For instance, five variables means $2^5 \times 5 = 160$ table entries, which is lengthy to write out and apply. A method suited for spreadsheet generation of a truth table that finds all the arrangements in order will be explained (Sullivan 2000). The customary appearance of the table is top row all T, bottom row all F; mixed rows are also in a conventional order. For

2-Variable Truth Table	
p	q
T	T
T	F
F	T
F	F

Figure 1. 2-variable truth table.

3-Variable Truth Table		
p	q	r
T	T	T
T	T	F
T	F	T
T	F	F
F	T	T
F	T	F
F	F	T
F	F	F

2-variable truth table

2-variable truth table

Figure 2. 3-variable truth table.

two variables p and q, the truth table is as shown in Figure 1. For three variables p, q, and r, the truth table is as shown in Figure 2.

The 2- and 3-variable truth tables reveal a repeating pattern. The pattern is that the entire 2-variable table is repeated twice (shaded portion) in the 3-variable table, once for p true and once for p false. Another way of describing the pattern is that there are $2^3 = 8$ rows of values for the 3-variable table. The first column headed p will get the first half of its rows *true* and the next half *false*. The 2-variable table values are supplied against the *trues* and against the *falses*, completing the table. The pattern with the 3-variable table will make the next higher table. The 4-variable table has 16 rows (2^4) of values and the first column headed p will get half the rows *true* and half *false*. The entire 3-variable table is supplied against the *trues* (light shading) and also against the *falses* (dark shading), which completes the 4-variable table (Figure 3).

4-Variable Truth Table			
p	**q**	**r**	**s**
T	T	T	T
T	T	T	F
T	T	F	T
T	T	F	F
T	F	T	T
T	F	T	F
T	F	F	T
T	F	F	F
F	T	T	T
F	T	T	F
F	T	F	T
F	T	F	F
F	F	T	T
F	F	T	F
F	F	F	T
F	F	F	F

3-variable truth table

3-variable truth table

Figure 3. 4-variable truth table.

The tables are rapidly made in a spreadsheet or word processor because both programs carry the copy and paste commands, which allow the highlighting of the section that is carried forward to the next table. The leftmost column will always be *trues* for the upper half of rows and *falses* for the bottom half of rows. Thus generating truth tables is advanced one at a time. Each table should be saved as a file, if it is not within a document already, so that it does not have to be regenerated in sequence each time it is needed. Saving a truth table as a Corel WordPerfect 8 file is done like saving any table. Click anywhere inside the table, then click on the menu names Edit|Select|Table|File|Save|Selected Text and supply a filename, e.g., "4var" for the 4-variable truth table. Making and saving in a spreadsheet will be explained next.

3. Spreadsheet Particulars

The point of having a truth table is to be able to evaluate expressions and arguments by the truth table method. The tabled arrangements so far seen contain no logical values for computing; they are made with a word processor and are just for looks. The spreadsheet does not recognize T and F as anything but labels. The logic values entered in a cell are the numbers 1, 0, or alternately the functions @TRUE, @FALSE. Either entry prints on the screen as 1, 0, but the editing window will show the @-function if it were actually entered. For speedy typing, 1 and 0 are preferred, but @-functions may clarify intent in some situations. Truth tables for computing are made with logic values in the way described in section 2. That way will be restated ahead in the context of spreadsheets (Underdahl 1994).

To know the best way to save and retrieve work, it is helpful to know a little more about the organization of spreadsheets. Spreadsheet programs are designed as notebooks of many tabbed pages, i.e., sheets. It is the whole notebook that is given a path name ending in a file name. To retrieve a particular page, the user selects the notebook name from the file menu and opens the sheet, say the one of truth tables, by clicking the mouse arrow on its tab. (Right clicking the tab allows the sheet to be named.) There is ample room on a sheet to fit all the truth tables needed. A needed table can be copied and pasted onto another sheet to start an actual problem. However, a better feature exists to retrieve a particular table.

In the Corel Quattro Pro 8 spreadsheet program, the feature to save and retrieve a large block of data is called a *block name*. A repertoire of, say, six (variable) truth tables is enough to meet most textbook problems. After a truth table is first made, the technique is: highlight it, click on the menu names Insert|Name|Cells, type an obvious name for the block and click "Add." For instance, the 2-variable truth table would be called by a brand new name "2logic," and so on. If they are forgotten, all of the names of blocks can be reviewed in the above menus. To put a named block on a new sheet, the user just types its name in a cell (the upper left corner of the block starts there) and the entire block of data is printed on the screen. The editing window will show that the cell contains the formula @ARRAY(2logic). (The block can be quickly deleted by just deleting the named [left top corner] cell.) Very large tables can be highlighted by the spreadsheet itself: click menu names View|Toolbars|Data Manipulation and click the SpeedSelect button, then proceed as previously outlined to name the block. A truth table can be built from the lower one by the copy and paste route, or by inserts of the block name of the lower table. It is important to note the following: (1) Accidentally deleting the sheet of truth tables or the name of a block will eliminate all or one of the block names and all problems that were started by typing a block-named truth table will have only the all-false instance and (2) The block names pertain only to that file in which they were made and saved.

The next higher table is made in the style that (T/F) truth tables were generated in section 2. For instance, by using copy and paste to create "3logic" skip over one cell from where the table is to start and type the previously named "2logic." The leftmost column is created by typing the number 1 or @TRUE at the first row, copying it, highlighting under it down to the last row of 2logic, and pasting, which fills the blank cells with logical trues ("1"). At the cell where the block should be repeated, "2logic" is typed again and entered. Completing the leftmost column begins by entering zero or @FALSE in the top cell, copying, highlighting down, and pasting to give the blank cells logical falses ("0"). With that, the logic table is complete and the block should be given its recognizable name.

4. Development

To apply the truth table to validating arguments, there first must be application to evaluating logical expressions. Then a rule for proving arguments must be given. Also the important logical connective of implication must be defined in truth table form. The method of showing invalidity in an argument (fallacies) will be given. Finally, some arguments in natural language will be symbolized and validated with truth tables.

4.1 Logical Expressions

A *logical expression* is any combination of the logical operators *and, or,* or *not,* which *implies* acting on logical variables. The *truth value* of the expression depends on the values that are tried from the truth table. Usually the truth value is a mixture of T and F, and the expression is said to be *contingent*. For example, the truth value of the logical expression *p and q* (symbolized $p \cdot q$) is true only if p and q are individually true, but is false for the other three arrangements of true and false. There are two special cases though, that involve implication, which are important in the proof of argument. An expression that evaluates as true for all table entries is a *tautology*; and one that evaluates as false for all table entries is a *contradiction.*

4.1.1 A Very Simple Example

We outline how to do a problem of two variables. On a new sheet "2logic" is typed into any cell, the truth table appears, and in the adjoining column is entered the 2-variable logical expression to be evaluated. The expression is copied down the rest of the rows, and the spreadsheet evaluates it. The 2logic table (Figure 4) reveals all possible values of the expression, here $p \cdot q$. As the table confirms, if either p or q or both are false, the conjunction is false.

The problem is begun at cell A1 by typing "2logic" which produces the block whose diagonal is A1..B4. The column identifications are typed in bold. Into cell C1 goes the spreadsheet rendition of the logical expression to be evaluated. So, in C1 ($p \cdot q$) is typed as +A1#AND#B1;

	A	B	C
	p	q	p · q
1	1	1	1
2	1	0	0
3	0	1	0
4	0	0	0

Note: Cell C1 formula can be written as +A1#AND#B1 or @AND(A1,B1).

Figure 4. A 2logic table.

the + sign prevents the formula from being mistaken and printed as a label. An equivalent formula is @AND(A1,B1). Note that the variable's cell location is used in the formula, and not the variable names p and q. The instruction cell and the empty column cells below that entry are highlighted and the Speedfill icon is clicked. The spreadsheet automatically adjusts the cell references, i.e., cell C2 holds +A2#AND#B2, and so on down to cell C4, and instantly computes and prints the values (ones and zeros) shown. This very simple example illustrates the truth table method of evaluating a logical expression.

4.1.2 Simple Expression

Some easy, successful examples of computing the logical value of expressions with a spreadsheet are shown for five examples involving two variables. The various connectives of the variables p and q are: p · q means p and q, ∨ means or, and ~ means not. The logic table that the spreadsheet uses is in columns A and B; five different expressions are in columns C–G (Figure 5). The computed results are easily mentally checked by the ordinary understanding of the logical results of combinations of true and false statements. Notice that all the expressions are contingencies.

	A	B	C	D	E	F	G
	p	q	p · q	p ∨ q	p · q ∨ q	~p	p · q ∨ ~p
1	1	1	1	1	1	0	1
2	1	0	0	1	0	0	0
3	0	1	0	1	1	1	1
4	0	0	0	0	0	1	1
	2logic table		five logical expressions evaluated				

Note: row 1 cell formulas are A1:@ARRAY(2logic) B1: 1
C1: +A1#AND#B1 or @AND(A1,B1)
D1: +A1#OR#B1 or @OR(A1,B1)
E1: +C1#OR#B1 or @OR(C1,B1)
F1: #NOT#A1
G1: +C1#OR#F1 or @OR(C1,F1

Figure 5. Example of logic table.

Because of a rule, compound expressions as in column G cannot be misread. The scope of a logical connective only extends to the next connective. So the dot includes q and stops there. If a different expression is meant, parentheses are used, e.g., $p \cdot (q \vee \sim p)$. For simplicity, 2-variable examples are shown, but a spreadsheet can compute the truth value of much more complicated expressions of numerous variables.

4.2 Implication Operation

Spreadsheets were never designed to perform symbolic logic, and they lack a built-in logical function for implication. Since many, many arguments contain the "if p, then q" construction of implication, the incomplete set of functions restricts symbolization. Fortunately, there is a "workaround" that fills in the set. The symbol of implication is a horseshoe between the p and q propositions, i.e., $p \supset q$. The propositions have names: p is the antecedent, and q is the consequent.

4.2.1 Truth Table for Implication

There is an insistence that, where p is true and q is false (row 2), the relation itself is false. If we decline to fix any more cases than that one, then column C shows the result. This resulting column for implication can be matched by column F for the expression $\sim (p \cdot \sim q)$ (Figure 6):

	A	B	C	D	E	F	G
	p	q	$p \supset q$	$\sim q$	$(p . \sim q)$	$\sim (p . \sim q)$	$\sim p \vee q$
1	T	T	T	F	F	T	T
2	T	F	F	T	T	F	F
3	F	T	T	F	F	T	T
4	F	F	T	F	F	T	T

Figure 6. Truth table for implication.

Expression F can be operated on to give the simpler expression shown in column G. Two substitutions transform the F to the G expression. Two needed expressions and their equivalents are

$\sim (p \cdot q) \equiv \sim p \vee \sim q$ De Morgan's law, and
$\sim \sim q \equiv q$ Double Negation.

These equivalents turn expression F into $\sim (p \cdot \sim q) \equiv (\sim p \vee \sim \sim q) \equiv (\sim p \vee q)$. The last expression becomes the working definition of implication:

$$\sim p \vee q \equiv p \supset q$$

When symbolizing implication $p \supset q$ in a spreadsheet, we will write P.IMP.Q as the column heading. The symbol set in Quattro Pro 8 does not include the horseshoe. We usually write the cell formula as #NOT#cell with p#OR#cell with q. An alternative formula is @OR(#NOT#cell with p, cell with q).

4.2.2 Paradoxes of Material Implication

The Figure 6 truth table at row 1 has both propositions p and q true and the implication also true. Row 2 is fixed by sensible insistence that truth implying falsehood should be a false relation. Rows 3 and 4 are surprising in that they both have the implication true even though one or both of the propositions is false. The truth values of the parts have become irrelevant to the truth value of implication itself. Rows 3 and 4 have been called "paradoxes of material implication." Suber (1997) has discussed implication with examples of propositions that show the truth table definition has reasonableness.

4.3 Proof of Arguments

An argument is a number of statements called premises that end in a final statement called the conclusion. No lines of an argument have to be objectively true. What matters is that the premises entail the conclusion, or better, that the conclusion logically follows from the premises. If so, the argument is *valid.* There are two ways to prove an argument is valid: by deductions, which are new statements justified by rules of inference that lead to the conclusion or by a truth table showing it is not invalid.

4.3.1 Controlling Rule

Notice that deductions prove validity, and truth tables prove noninvalidity. An invalid argument cannot be shown so by deductions, but is readily detectable with a truth table. A truth table demonstration of validity uses a reverse approach, because the controlling rule for validity is essentially a negative one. The rule is: *a valid argument cannot have true premises and a false conclusion* ("truth can't come from a lie"). We look only for that forbidden case in a truth table, and if it does not appear, the argument is valid.

Let us prove a very simple argument both ways:

1.	p	premise 1
2.	q	premise 2
$\therefore$	$p \cdot q$	conclusion

Notice that the truth table in section 4.1.1 contained columns of these expressions, but there was no mention that the expressions could be united to make an argument.

This argument is valid on sight by Conjunction, rule of inference no. 8 of Copi (1965).

The truth table is as shown in Figure 7.

	A	B	C
	p	q	$p \cdot q$
1	T	T	T
2	T	F	F
3	F	T	F
4	F	F	F

Figure 7. Truth table.

The premises 1 and 2 are respective headings for columns A and B; the conclusion heads column C. Only in row 1 do both premises have the logic value of true. We do not need to further consider the other rows. Back in row 1, the conclusion is true, which is to say it is not false. So the forbidden case (true premises and a false conclusion) does not arise in this argument, and it is valid.

4.3.2 Configuring Arguments for Spreadsheets

A small truth table can be visually inspected for the invalidating case, but not a large one. The premises and conclusion are scattered, and there are too many rows. There are reconfigurations of an argument that make for automatic checking for validity. The idea is that in a valid argument nothing new is said by the conclusion—the premises imply it. Further, in the only meaningful case the premises are individually true, so conjoining them into one expression leaves the logic value as true. Let the conjoined premises be the antecedent of an implication, and let the conclusion be the consequent. A complicated argument is reduced this way to an expression of $p \supset q$. Recall the truth table for implication:

	A	B	C
	p	q	$p \supset q$
1	T	T	T
2	T	F	F
3	F	T	T
4	F	F	T

where valid argument in this implication configuration will always show trues because case 2 is not present by definition of validity. In this configuration, the argument has been recast into a tautologous expression. In spreadsheet form, the valid argument (tautology form) has all ones in the implication column; the invalid argument holds a zero wherever row 2 (shaded) comes into play.

Now consider the argument reconfigured as $p \cdot \sim q$, where as before p is the conjoined premises and q is the conclusion. Therefore, the truth table is

	A	B	C
	p	q	$p \cdot \sim q$
1	T	T	F
2	T	F	T
3	F	T	F
4	F	F	F

A valid argument will always show falses in this configuration. The argument has been recast into a contradictory expression. In spreadsheet form, the valid argument (contradiction form) has all zeros in the $p \cdot \sim q$ column; the invalid argument holds a one wherever row 2 (shaded) comes into play.

In spreadsheet evaluation of arguments, we will always make final columns where the argument is in tautology and contradiction form. (One or the other would suffice.) If it is valid, a glance down the columns will show that every row is one and zero, respectively. If it is invalid, each column will be spoiled by a discordant, opposite symbol.

4.3.3 Restyled Truth Table for an Argument

Take the same argument as before, but improve it by identifying the columns (Figure 8):

	Premise 1	Premise 2	Conclusion
	A	B	C
	p	q	$p \cdot q$
1	T	T	T
2	T	F	F
3	F	T	F
4	F	F	F

Figure 8. Improved truth table.

See how it looks in a restyled truth table (Figure 9):

	Premise 1	Premise 2	Conclusion	Tautology	Contradiction
	A	B	C	D	E
	p	q	$p \cdot q$	$p \cdot q \supset p \cdot q$	$(p \cdot q) \cdot \sim(p \cdot q)$
1	T	T	T	T	F
2	T	F	F	T	F
3	F	T	F	T	F
4	F	F	F	T	F

Figure 9. Restyled truth table.

The first style (Figure 8) shows validity by inspection. Only in row 1 are the premises both true, and looking at the conclusion in column C, it is not false, so the controlling rule has not been violated. The second style (Figure 9) added columns D and E to reconfigure the argument as a tautology and a contradiction, respectively. Specifically, column D shows an implication having the conjoined premises as the antecedent and the conclusion as the consequent. That is the prescription for testing that an argument is a tautology, hence valid. If the argument is valid, the tautology necessarily will show a column of all trues (shaded area), and the contradiction will show the opposite. Column E conjoins the premises and the negation of the conclusion. Either column D or E suffices, but both are given to illustrate the method. Validity is easier to spot in the new style, and it is a style adaptable to spreadsheet calculation.

4.4 Fallacies

An invalid argument is called a *fallacy*. Some arguments, anciently known to be invalid, recur so often that they have been named. One called Affirming the Consequent goes

Affirming the Consequent
1. $p \supset q$
2. q

$\therefore p$ (Invalid argument)

Examining the argument with a truth table shows that it violates the controlling rule, and thus is invalid (Figure 10).

	A	B	C
	p	q	$p \supset q$
1	T	T	T
2	T	F	F
3	F	T	T
4	F	F	T

Figure 10. Affirming the Consequent truth table.

The premises 1 and 2 are the headings of columns C and B, respectively. The premises are both true only on rows 1 and 3. The conclusion, heading column A, is true on row 1, but is false on row 3. In other words, row 3 shows an instance of true premises and a false conclusion (shaded area), and that makes the argument itself invalid. The other rows (2 and 4) are irrelevant to the consideration, since the premises are not both true there. The invalid argument resembles a valid argument called *Modus Ponens,* listed in section 4.5.

When the truth table adds tautology and contradiction columns (E and F) of section 4.3.2, the argument's invalidity is observed differently, as shown in Figure 11. Since E and F do not contain just one truth value (row 3), the argument is shown to be invalid. Note that to save space the long expression in column D is called D (it is the conjoined premises) when used in columns

	A	B	C	D	E	F
	p	q	$p \supset q$	$(p \supset q) \cdot q$	$D \supset p$	$D \cdot {\sim}p$
1	T	T	T	T	T	F
2	T	F	F	F	T	F
3	F	T	T	T	F	T
4	F	F	T	F	T	F

Figure 11. Truth table with tautology and contradiction columns.

E and F. This same space saving is done ahead when spreadsheet headings are made. There should not be any confusion that the column letter is merely standing for the expression heading the column.

Another named fallacy is Denying the Antecedent (Figure 12).

Denying the Antecedent
1. $p \supset q$
2. ${\sim}p$
$\therefore {\sim}q$ (Invalid argument)

	A	B	C	D	E
	p	q	${\sim}p$	${\sim}q$	$p \supset q$
1	T	T	F	F	T
2	T	F	F	T	F
3	F	T	T	F	T
4	F	F	T	T	T

Figure 12. Denying the Antecedent truth table.

The premises 1 and 2 are the headings of columns E and C, respectively. The premises are both true only on rows 3 and 4. The conclusion, heading column D, is true on row 4, but is false on row 3. In other words, row 3 shows an instance of true premises and a false conclusion (shaded area), and that makes the argument itself invalid. The other rows (1 and 2) are irrelevant to the consideration, since the premises are not both true there. This invalid argument resembles a valid argument called *Modus Tollens*, listed in section 4.5.

There is an oddity among fallacies. The fallacy known as the Slippery Slope, which is quite popular with television political pundits, is based on a valid argument called the Hypothetical Syllogism. The valid argument, is used in section 4.4.2, and goes

Hypothetical Syllogism
1. $p \supset q$
2. $q \supset r$
$\therefore p \supset r$ (Valid argument)

The Hypothetical Syllogism has the consequent of one implication becoming the antecedent of the next. The chain can go on and on, past the two premises shown here, but the conclusion is always an implication having the first antecedent and the last consequent of the chain. In popular talk, the valid argument transmogrifies into the fallacy. The full package is a warning that if we do this first semi-innocuous thing then it will lead to the next thing, which will lead to the next, etc., and finally we are given a consequence that everyone admits is dire. So, we ought never to do the first thing. The fallacy is the speaker's omniscient presumption in stating that one thing always causes another, and that the next change is irresistible or inevitable, and that events can be forecast to an ordained end. Usually though the speaker is too lazy to think of a chain of worsening developments, and the speech shortens from fallacy to metaphor. "If we do that, we're on a slippery slope." So, a fallacy is converted from invalid argument to a "puts-a-stop-to-it" reason. So besides the oddity of a valid form being converted to a fallacy, a double oddity occurs, e.g., the fallacy itself is invoked to prove the argument. One more way this fallacy is shuffled off on people is the even shorter plaint, "If we do this, where will it end?"

Very few invalid arguments are named. Most fallacies are not based on logic at all (informal fallacies), but are really techniques of propaganda, persuasion, advocacy, and grifting. The aim in using a fallacious argument is overcoming someone. A fallacious argument cannot for the most part be analyzed by a truth table. The number of catalogued fallacies is three dozen, and Web sites describing them can be found by a search engine looking for "fallacy."

4.5 Examples of Arguments

The following subsections give three examples of verbal arguments. They are first translated into symbols, and then, to contrast the methods, are proven by the rules of inference, and then proven using the truth table method on a spreadsheet. The arguments are posed in very stilted English, with all premises explicit, and the "if...then" construction used for implication. The awkwardness comes from the strict argumentation, which is probably there for pedagogical reasons. Spoken English sometimes leaves out structure, and the auditor himself must supply lines that the speaker thinks are unnecessary for comprehension. Additionally many arguments are inherently complex. So for several reasons, symbolizing can get difficult.

A proof proceeds by lengthening the symbolized argument with new lines justified by the rules of inference. These rules give equivalents that replace all or part of the premises and subarguments that combine lines to yield a new line. The proof ends when a line reaches the argument's conclusion. In all, Copi (1965) provides 19 rules (10 equivalents plus 9 subarguments) and two techniques, conditional proof and indirect proof. We will not use examples needing the two techniques. As for the rules themselves, they are recommended as practical; they are not independent, meaning that some could be disposed of and proofs still accomplished. Retaining more rules than are strictly needed, is useful.

The subarguments used in the examples have the following forms. They can all be proven by a truth table:

Modus Ponens	*Modus Tollens*	Destructive Dilemma
1. $p \supset q$	1. $p \supset q$	1. $(p \supset q) \cdot (r \supset s)$
2. p	2. $\sim q$	2. $\sim q \vee \sim s$
$\therefore q$	$\therefore \sim p$	$\therefore \sim p \vee \sim r$

Disjunctive Syllogism	Hypothetical Syllogism
1. $p \vee q$	1. $p \supset q$
2. $\sim p$	2. $q \supset r$
$\therefore q$	$\therefore p \supset r$

The only nonobvious equivalents used are De Morgan's laws and Exportation. Both may be proven by truth tables:

De Morgan's laws	Exportation
$\sim(p \vee q) \equiv \sim p \cdot \sim q$	$p \supset (q \supset r) \equiv (p \cdot q) \supset r$
$\sim(p \cdot q) \equiv \sim p \vee \sim q$	

4.5.1 Three Symbol Argument

According to Copi (1965) problem II-4, p. 48:

> If he uses good bait then if the fish are biting then he catches the legal limit. He uses good bait but he does not catch the legal limit. Therefore, the fish are not biting. (G, B, C):

1. $G \supset (B \supset C)$	
2. $G \cdot \sim C$	$/ \therefore \sim B$
3. $(G \cdot B) \supset C$	1, Exportation
4. $\sim C \cdot G$	2, Commutation
5. $\sim C$	4, Simplification
6. $\sim(G \cdot B)$	3, 5, *Modus Tollens*
7. $\sim G \vee \sim B$	6, De Morgan
8. G	2, Simplification
9. $\sim B$	7, 8, Disjunctive Syllogism

Line 3 replaces line 1 with its equivalent by a rule called Exportation. Lines 4 and 5 play with line 2, i.e., if we have two things together then we have one of them ($\sim C$) by itself. An ancient argument form known as *Modus Tollens* justifies line 6. This argument has the form 1. $p \supset q$ 2. $\sim q / \therefore \sim p$. It is clearly seen that lines 3, 5, and 6 have the form of *Modus Tollens*. Visual recognition of argument forms is a standard procedure in applying the rules of inference. Line 7 is equivalent to line 6 by one of De Morgan's laws. Lines 7 and 8 form the premises of another argument form known as Disjunctive Syllogism, whose conclusion is line 9. But line 9 is also the conclusion of the argument in question, which is proven valid. (As stated in section 4.5, the three named subarguments can themselves be proven valid by truth tables, which we shall not bother to give.)

Since Copi's rules of inference are more than sufficient to prove arguments, more than one proof may sometimes be found. A shorter proof goes

1. $G \supset (B \supset C)$	
2. $G \cdot \sim C$	$/ \therefore \sim B$
3. G	2, Simplification
4. $B \supset C$	1, 3, *Modus Ponens*
5. $\sim C \cdot G$	2, Commutation
6. $\sim C$	5, Simplification
7. $\sim B$	4, 6, *Modus Tollens*

The truth table method is shown in Figure 13. The spreadsheet page is titled with the problem number, and the block of work is located in a frame of rows (4–13) and columns (B–K). Cells B5..D5 are titled by the variables of the problem (G, B, C). In cell B6, typing the previously made and saved block name "3logic" generates the 3-variable table. Important columns that identify premises, conclusion, tautology, and contradiction are so labeled. Other columns are workups to the needed premises and conclusion. Column I is the conjunction of both premises, and it is the antecedent of the implication in column J. The consequent of the implication is the conclusion in column E. As section 4.1.2 states, this arrangement of a valid argument is a tautology. Column K is dispensable, and just shows the alternative arrangement that gives a contradiction.

The necessary cell formulas are entered on row 6. The cell formulas in the first row are printed below the block of work. It is unnecessary to print the other formulas as the spreadsheet automatically adjusts the cell formula down each column. For instance, cell E6 contains the formula #NOT#C6 and beneath it is cell E7, which must contain the formula #NOT#C7. A common error in writing formulas is confusing the column letter with the column heading. For instance, in column E headed by $\sim B$ the formula in cell E6 must refer to the contents in cell C6 and be written as #NOT#C6, not as #NOT#B6. Another thing to notice is that formulas can be written in other ways. The implication in column G is straightforwardly written as material implication is symbolized, i.e., $p \supset q \equiv \sim p \vee q$. But in column J, we choose to use the @OR-form. Another kind of choice is displayed in column K where an @AND-form is preferred for readability over the form needing adjacent pound signs, i.e., +I6#AND##NOT#E6.

Copi (1965) Problem II-4, p. 48

II 4	B	C	D	E	F	G	H	I	J	K
4				CONCLUSION		PREM1	PREM2	PREM12	TAUTOLOGY	CONTRADICTION
5	G	B	C	~B	B.IMP.C	G.IMP.F	G & ~C	GH	I.IMP.~B	I & ~(~B)
6	1	1	1	0	1	1	0	0	1	0
7	1	1	0	0	0	0	1	0	1	0
8	1	0	1	1	1	1	0	0	1	0
9	1	0	0	1	1	1	1	1	1	0
10	0	1	1	0	1	1	0	0	1	0
11	0	1	0	0	0	1	0	0	1	0
12	0	0	1	1	1	1	0	0	1	0
13	0	0	0	1	1	1	0	0	1	0

Cell Formulas

```
II 4:B6:   @ARRAY(3logic)
II 4:C6:   1
II 4:D6:   1
II 4:E6:   #NOT#C6
II 4:F6:   #NOT#C6#OR#D6
II 4:G6:   #NOT#B6#OR#F6
II 4:H6:   @AND(B6,#NOT#D6)
II 4:I6:   @AND(G6,H6)
II 4:J6:   @OR(#NOT#I6,E6)
II 4:K6:   @AND(I6,#NOT#E6)
```

Figure 13. Three symbol argument truth table method.

Printing is adjusted as desired from File|Print|Page Setup to tabs controlling details, such as orientation, framework, centering, heading, and many others. Print Preview shows the adjustments on screen. When the setup is sufficiently close to desired, the tab Named Settings allows the setup to be saved and reused by name, e.g., II 4. Any further changes are saved with buttons Update and OK.

4.5.2 Four Symbol Argument

According to Copi (1965) problem II-2, p. 48:

> If the supply of silver remains constant and the use of silver increases then the price of silver rises. If an increase in the use of silver implies that the price of silver rises then there will be a windfall for speculators. The supply of silver remains constant. Therefore, there will be a windfall for speculators.
> (S, U, P, W):

1. $(S \cdot U) \supset P$
2. $(U \supset P) \supset W$
3. S $/ \therefore W$
4. $S \supset (U \supset P)$ 1, Exportation
5. $S \supset W$ 4, 2, Hypothetical Syllogism
6. W 5, 3, *Modus Ponens*

Lines 4–6 are deductions made by replacing a premise with its equivalent or by combining two lines via a recognized argument. Line 4 replaces line 1 with its equivalent, and names the rule by which it is permitted. Lines 4 and 2 are in the form of Hypothetical Syllogism, whose conclusion is line 5. Lines 5 and 3 produce line 6, the conclusion of a different subargument *Modus Ponens*. But line 6 is also the conclusion to be proved, so the argument is valid by means of (Copi's) rules of inference.

A proof doesn't have to be unique. Two applications of *Modus Ponens* will also get to the conclusion.

1. $(S \cdot U) \supset P$
2. $(U \supset P) \supset W$
3. S $/ \therefore W$
4. $S \supset (U \supset P)$ 1, Exportation
5. $U \supset P$ 4, 3, *Modus Ponens*
6. W 2, 5, *Modus Ponens*

The truth table method is shown in Figure 14. The spreadsheet page is titled with the problem number and the block of work is located in a frame of rows (3–20) and columns (B–L). Cells B4..E4 are titled by the variables of the problem (S, U, P, W). In cell B5 typing "4logic" generates the 4-variable table. Important columns that identify premises, conclusion, tautology, and contradiction are so labeled. Other columns are workups to the needed premises and conclusion. Column J is the conjunction of all three premises, and the antecedent of the implication in column K. The consequent of the implication is the conclusion in column E. As section 4.1.2 states, this arrangement (in column K) of a valid argument is a tautology. Column L is dispensable, and just shows the alternative arrangement that gives a contradiction.

The necessary cell formulas are entered on row 5. Filling the rest of a column can be done two ways. For large tables, a method in section 4.4.3 is better. For small tables the following method works well. When the row is completed, one practice is to highlight a column from row 5 to row 20, and hit Speedfill. The cells are automatically filled with the formula adjusted for the cell location. It is good practice to put the spreadsheet in manual recalculation mode, which prevents it operating every time a cell is changed. The sequence needed is: Format|Notebook Recalculate Tab|Manual. Pressing key F9 or a calculator icon on the bottom bar of the screen triggers the calculation. Forgetting to do this step can lead to hunting for a formula mistake.

Copi (1965) Problem II-2, p. 48

II 2	B	C	D	E	F	G	H	I	J	K	L
3	PREM3			CONCLUSION		PREM1		PREM2	PREM123	TAUTOLOGY	CONTRADICTION
4	S	U	P	W	SU	SU.IMP.P	U.IMP.P	H.IMP.W	GIB	J.IMP.W	J&~W
5	1	1	1	1	1	1	1	1	1	1	0
6	1	1	1	0	1	1	1	0	0	1	0
7	1	1	0	1	1	0	0	1	0	1	0
8	1	1	0	0	1	0	0	1	0	1	0
9	1	0	1	1	0	1	1	1	1	1	0
10	1	0	1	0	0	1	1	0	0	1	0
11	1	0	0	1	0	1	1	1	1	1	0
12	1	0	0	0	0	1	1	0	0	1	0
13	0	1	1	1	0	1	1	1	0	1	0
14	0	1	1	0	0	1	1	0	0	1	0
15	0	1	0	1	0	1	0	1	0	1	0
16	0	1	0	0	0	1	0	1	0	1	0
17	0	0	1	1	0	1	1	1	0	1	0
18	0	0	1	0	0	1	1	0	0	1	0
19	0	0	0	1	0	1	1	1	0	1	0
20	0	0	0	0	0	1	1	0	0	1	0

Cell Formulas

```
II 2:B5:    @ARRAY(4logic)
II 2:C5:    1
II 2:D5:    1
II 2:E5:    1
II 2:F5:    +B5#AND#C5
II 2:G5:    #NOT#F5#OR#D5
II 2:H5:    #NOT#C5#OR#D5
II 2:I5:    #NOT#H5#OR#E5
II 2:J5:    @AND(G5,I5,B5)
II 2:K5:    #NOT#J5#OR#E5
II 2:L5:    @AND(J5,#NOT#E5)
```

Figure 14. Four symbol argument truth table method.

After key F9 is pressed, the block of work shows the printed values of Figure 14. A glance at column K shows that the tautology has all ones, as it must for a valid argument (or if preferred, column L has all zeros).

4.5.3 Five Symbol Argument

According to Copi (1965) problem II-7, p. 48:

> If he attracts the farm vote then he will carry the rural areas, and if he attracts the labor vote then he will carry the urban centers. If he carries both the urban centers and the rural areas then he is certain to be elected. He is not certain to be elected. Therefore, either he does not attract the farm vote or he does not attract the labor vote. (F, R, L, U, C):

1.	$(F \supset R) \cdot (L \supset U)$	
2.	$(U \cdot R) \supset C$	
3.	$\sim C$	$/ \therefore \sim F \vee \sim L$
4.	$\sim (U \cdot R)$	2, 3, *Modus Tollens*
5.	$\sim U \vee \sim R$	4, De Morgan
6.	$\sim R \vee \sim U$	5, Commutation
7.	$\sim F \vee \sim L$	1, 6, Destructive Dilemma

Lines 2 and 3 have the form of the premises of *Modus Tollens,* so we can infer the conclusion of that argument which here takes the form of line 4. Line 5 is the equivalent of line 4 by one of De Morgan's laws. Line 6 simply switches the order of terms in line 5. Lines 1 and 6 have the form of the premises of another argument called the Destructive Dilemma, so we can infer the conclusion of that argument which here takes the form of line 7. But line 7 is also the conclusion of the argument in question, which is proved valid.

The truth table method is described next. The spreadsheet page (Figure 15) is titled with the problem number and the block of work is located in a frame of rows (3–36) and columns (B–P). Cells B4..F4 are titled by the variables of the problem (F, R, L, U, C). In cell B5 typing "5logic" generates the 5-variable table. Important columns that identify premises, conclusion, tautology, and contradiction are so labeled. Other columns are workups to the needed premises and conclusion. Column M is the conjunction of all three premises, and the antecedent of the implication in column O. The consequent of the implication is the conclusion in column N. As section 4.1.2 states, this arrangement (in column O) of a valid argument is a tautology. Column P is dispensable, and just shows the alternative arrangement that gives a contradiction.

The necessary cell formulas are entered on row 5. The cell formulas in the first row are printed below the block of work. Filling the long columns can be done differently than using Speedfill, as was done in section 4.4.1. The method uses array formulas and works mainly on the first row with manual recalculation off. Array formulas specify that arithmetic or logical operations be performed throughout an entire block. For instance, in cell G5 we want the implication operation performed like this: $F \supset R$ or #NOT#B5#OR#C5. But since we really want the column filled down to row 36, we specify a range of cells by the double dot notation #NOT#B5..B36#OR#C5..C36. The spreadsheet supplies the @ARRAY prefix if we don't care to type it ourselves. The formula is executed pairwise from B5 and C5 together through to B36 and C36. In other words, after the formula is correctly entered the spreadsheet properly fills column G from row 5 down to row 36. The advantage of the array formula is that we don't have to scroll down a long distance. Columns H..L of row 5 have been filled via the array command. In cell M5 we come to the step of combining all the premises into one expression. Since there are three premises, it is shorter to use an @AND command with commas, rather than repeating the #AND# command. At M5 though we have reverted to Speedfill to fill the work block.

Copi (1965) Problem II-7, p. 48

II 7	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
3								PREM1		PREM2	PREM3	PREM123	CONCLUSION	TAUTOLOGY	CONTRADICTION
4	F	R	L	U	C	F.IMP.R	L.IMP.U	GH	UR	J.IMP.C	~C	IKL	~F v ~L	M.IMP.N	M & ~N
5	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0
6	1	1	1	1	0	1	1	1	1	0	1	0	0	1	0
7	1	1	1	0	1	1	0	0	0	1	0	0	0	1	0
8	1	1	1	0	0	1	0	0	0	1	1	0	0	1	0
9	1	1	0	1	1	1	1	1	1	1	0	0	1	1	0
10	1	1	0	1	0	1	1	1	1	0	1	0	1	1	0
11	1	1	0	0	1	1	1	1	0	1	0	0	1	1	0
12	1	1	0	0	0	1	1	1	0	1	1	1	1	1	0
13	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0
14	1	0	1	1	0	0	1	0	0	1	1	0	0	1	0
15	1	0	1	0	1	0	0	0	0	1	0	0	0	1	0
16	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0
17	1	0	0	1	1	0	1	0	0	1	0	0	1	1	0
18	1	0	0	1	0	0	1	0	0	1	1	0	1	1	0
19	1	0	0	0	1	0	1	0	0	1	0	0	1	1	0
20	1	0	0	0	0	0	1	0	0	1	1	0	1	1	0
21	0	1	1	1	1	1	1	1	1	1	0	0	1	1	0
22	0	1	1	1	0	1	1	1	1	0	1	0	1	1	0
23	0	1	1	0	1	1	0	0	0	1	0	0	1	1	0
24	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0
25	0	1	0	1	1	1	1	1	1	1	0	0	1	1	0
26	0	1	0	1	0	1	1	1	1	0	1	0	1	1	0
27	0	1	0	0	1	1	1	1	0	1	0	0	1	1	0
28	0	1	0	0	0	1	1	1	0	1	1	1	1	1	0
29	0	0	1	1	1	1	1	1	0	1	0	0	1	1	0
30	0	0	1	1	0	1	1	1	0	1	1	1	1	1	0
31	0	0	1	0	1	1	0	0	0	1	0	0	1	1	0
32	0	0	1	0	0	1	0	0	0	1	1	0	1	1	0
33	0	0	0	1	1	1	1	1	0	1	0	0	1	1	0
34	0	0	0	1	0	1	1	1	0	1	1	1	1	1	0
35	0	0	0	0	1	1	1	1	0	1	0	0	1	1	0
36	0	0	0	0	0	1	1	1	0	1	1	1	1	1	0

Cell Formulas

```
II 7:B5:    @ARRAY(5logic)
II 7:C5:    1
II 7:D5:    1
II 7:E5:    1
II 7:F5:    1
II 7:G5:    @ARRAY(#NOT#B5..B36#OR#C5..C36)
II 7:H5:    @ARRAY(#NOT#D5..D36#OR#E5..E36)
II 7:I5:    @ARRAY(G5..G36#AND#H5..H36)
II 7:J5:    @ARRAY(E5..E36#AND#C5..C36)
II 7:K5:    @ARRAY(#NOT#J5..J36#OR#F5..F36)
II 7:L5:    @ARRAY(#NOT#F5..F36)
II 7:M5:    @AND(I5,K5,L5)
II 7:N5:    @OR(#NOT#B5,#NOT#D5)
II 7:O5:    #NOT#M5#OR#N5
II 7:P5:    @AND(M5,#NOT#N5)
```

Figure 15. Five symbol argument truth table method.

At cells M5, N5, O5, and P5, we write the formulas without the array command and see that the top row is properly one or zero in the tautology and contradiction cell, respectively. (If they weren't, we could go back and correct a mistake in the formulas.) Then with one highlight of the block (M5..P36), Speedfill creates the filling.

5. Discussion

Limitations of this material and fields related to it are discussed next.

Logic outcomes (true/false) are recognized by spreadsheets in order to permit simple comparisons and cause something else to be done. The function used for these tests is an @IF-function, which usually employs inequality tests to reach a logic outcome. The explicit logic operators *and, or, not* are seldom used, but their opportune presence leads to a symbolic logic capability in spreadsheets.

To symbolize more complicated sentences than seen in the examples of section 4.5, additional rules have to be developed. The greatest extension is caused by quantification logic, which is needed to deal with sets of objects and arguments about them. For instance, symbolizing *All dogs are mammals* and *Not all mammals are dogs* requires new notation that looks like calculus and not like the bare symbols in the examples. Relations extend quantification even more and allow symbolizing sentences like *Grant and Lincoln were acquainted.* Copi (1965) explains quantifiers in chapter 4 and relations in chapter 5. Truth tables are inapplicable to deciding validity of arguments involving those logics. Furthermore this report does not use other definitions of implication (strict implication), which try to avoid the paradoxes of material implication, section 4.2.2. Also while we only admit two truth values (true and false), multi-valued logic and fuzzy logic are accepted elsewhere. To sum up, there are these restrictions on the use of truth tables:

- No sentences but simple declaratives or conditionals (implication).
- No other than material implication is used in the report.
- No quantifier logic (nouns modified by words like *all, some, only, every, each*).
- No relational logic (phrases like *x belongs to y, x is acquainted with y, x is related to y*).
- No multi-value logic; logical variables can have more than two values.

Aside from logic, truth tables make an appearance in probability and electrical engineering.

They are a visual aid to solving some basic probability problems. In probability, coin-flipping problems are encountered. The possible outcomes of a single coin flip are heads and tails, which correspond to the true and false values of a logical variable. The four arrangements of heads and tails from flipping two coins correspond to the two-variable table, and so on. Flipping just $n = 3$ coins can create confusion as there are 8 arrangements (2^n) of heads and tails (rows of a truth table) and 4 outcomes ($n + 1$), i.e., no head, one head, two heads, three heads. Generating this situation with the "3var" truth table allows one to answer numerous small questions about the probability of various numbers of heads (or tails) appearing. For instance, counting how many arrangements there are of each outcome leads to the probability of each outcome. Misunderstandings in conditional probability are cleared up with a table. The visual solution of a truth table aids understanding of the shorter solution by a formula (Runyon and Haber 1971).

In electrical engineering, the logic circuit course introduces a truth table and meshes it with Boolean algebra to create formulas that are interpreted as switching circuits. Their truth table is inverted: top row is all zeros (all switches open); the bottom row is all ones (all switches closed) (Whitesitt 1995). It is also shown that to each truth table corresponds a Boolean expression and vice versa. So a switch circuit represents a truth table and a Boolean expression and vice versa. The simplest examples of this are $p \cdot q$, which stands for two switches in series, and $p \vee q$, which stands for two switches in parallel. Sloane (1996) provides an example of going from a truth table to a Boolean expression. As the Boolean expressions tend to be long sums, simplification of an expression with the axioms of Boolean algebra has to be mastered. The last step corresponds to finding a simpler (cheaper to build) circuit that obeys the truth table.

6. Conclusion

Though not by design, spreadsheets can be applied to solve some problems in symbolic logic. The spreadsheet can evaluate a logical expression, mixtures of logical connectives and symbols. It can generate a truth table, an exhaustive array of true and false values for a useful number of variables. It can display a truth table with headings of logical expressions. The expressions can be premises and conclusion of an argument. The truth table can be inspected to show that the argument is valid. Validity occurs if there is no row of the truth table where the premises evaluate as true and the conclusion evaluates as false. If the argument is reconfigured as an implication, the spreadsheet can automatically searched for the invalidating case. The reconfiguration makes a valid argument into a tautologous expression. To do this, premises are joined by the *and* connective, and the conjunction become the if-proposition of an implication. The conclusion is made the then-proposition of the implication. Only for a valid argument does the implication column of the spreadsheet show all ones.

7. References

Copi, I. M. *Symbolic Logic*. Second edition, New York: Macmillan, 1965.

Runyon, R. P., and A. Haber. *Fundamentals of Behavioral Statistics*. Second edition, Reading, MA: Addison-Wesley, p. 161, 1971.

Sloane, T. "Extracting a Boolean Function From a Truth Table." <http://www.cs.jcu.edu.au/ftp/web/teaching/Subjects/cp1200/1996/org/node24.html>, James Cook University, Queensland, Australia, 1996.

Suber, P. "Paradoxes of Material Implication."<http://www.earlham.edu/~peters/courses/log/mat-imp.htm>, Earlham College, Richmond, IN, 1997.

Sullivan, J. D. "Spreadsheet Generation of a Truth Table." ARL-MR-497, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, September 2000.

Underdahl, B. *Using Quattro Pro 6 for Windows, Special Edition*. Indianapolis, IN: Que Corp., 1994.

Whitesitt, J. E. *Boolean Algebra and Its Applications*. Mineola, NY: Dover, 1995.

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2002	Final, January 2000–June 2002

4. TITLE AND SUBTITLE
Spreadsheet Proof of Arguments

5. FUNDING NUMBERS
28T8FC

6. AUTHOR(S)
John D. Sullivan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-WM-TB
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-2815

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes the use of a spreadsheet to solve simple problems in prepositional logic. A spreadsheet can easily generate and recall truth tables, and with its built-in logical functions and connectives it can calculate the truth value of logic expressions. Expressions can also be premises and together with a conclusion constitute an argument, which can be shown valid with a truth table.

This report covers the application of a spreadsheet to truth table generation, evaluation of logical expressions, recasting arguments into spreadsheet form, and demonstration of validity.

14. SUBJECT TERMS
truth table, logic, spreadsheet, Quattro Pro

15. NUMBER OF PAGES
29

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.